

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-078894

(43)Date of publication of application : 24.03.1998

(51)Int.Cl.

G06F 11/30

G06F 11/22

(21)Application number : 08-252487

(71)Applicant : HITACHI LTD

(22)Date of filing : 03.09.1996

(72)Inventor : KUNINISHI MOTOHIDE

KATO HIROSHI

ITO TSUTOMU

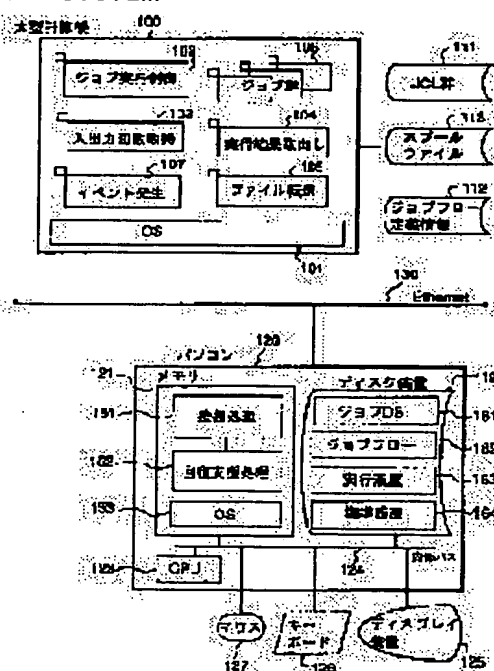
HIROZAWA TOSHIO

(54) METHOD FOR SUPPORTING FAILURE RESTORATION FOR COMPUTER SYSTEM

(57)Abstract :

PROBLEM TO BE SOLVED: To support the restoration of failure by shortening a restoring time and finding a job being the cause of the failure at the time of the generation of the failure in a computer system.

SOLUTION: At the time of starting or ending the execution of a job in job execution control 102, or at the time of an input and output information capture processing 103 provided in the close processing of a file input and output processing, an event generation processing 107 is called, a job start event, job end event, and close event are generated, and an execution history 163 constituted of an end code, starting time, ending time, executing time, and input and output history or the like is prepared for each job based on those events by a monitor processing 151. Then, at the time of the generation of failure, a failure supporting processing 152 is activated, the execution history 163 is compared with a standard history 164 based on normal execution, the abnormality of the number of times of input and output, executing time, and end code is checked, an abnormality flag is stored in the execution history, and a job in which the abnormality is generated is defined as a job being the cause of the failure. Then, after the restoration and re-assignment of a file, a re-executing instruction is issued.



LEGAL STATUS

[Date of request for examination] 08.03.2001

[Date of sending the examiner's decision of rejection] 19.11.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

BEST AVAILABLE COPY

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-78894

(43)公開日 平成10年(1998) 3月24日

(51)Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/30	3 0 5		G 0 6 F 11/30	3 0 5 D
11/22	3 6 0		11/22	3 6 0 E

審査請求 未請求 請求項の数 8 F D (全 11 頁)

(21)出願番号 特願平8-252487

(22)出願日 平成8年(1996)9月3日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 国西 元英

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 加藤 拓

神奈川県川崎市幸区鹿島田890番地 株式

会社日立製作所情報システム事業部内

(72)発明者 伊藤 勉

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 笹岡 茂 (外1名)

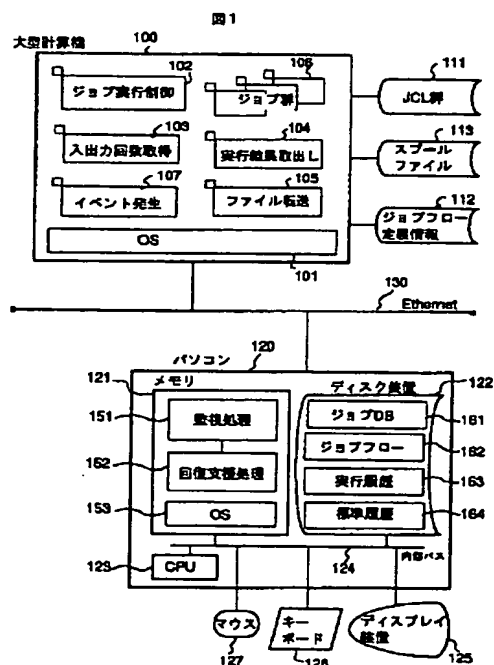
最終頁に続く

(54)【発明の名称】 計算機システムの障害回復支援方法

(57)【要約】

【課題】 計算機システムにおける障害の発生に対し、回復時間の短縮を狙い、障害の原因となったジョブを見つけ出し、障害回復の支援を行なう。

【解決手段】 ジョブ実行制御102におけるジョブの実行開始時、実行終了時、またファイル入出力処理のクローズ処理に設けられた入出力情報取得処理103時にイベント発生処理107を呼出し、ジョブ開始イベント、ジョブ終了イベント、クローズイベントを発生させ、監視処理151はこれらイベントに基づき各ジョブにつき終了コード、開始時刻、終了時刻、実行時間、入出力履歴等からなる実行履歴163を作成し、障害発生時に障害支援処理152を起動し、実行履歴163と正常な実行に基づく標準履歴164を比較し、入出力回数、実行時間、終了コードの異常を調べ、異常フラグを実行履歴に格納し、異常の発生したジョブを障害原因ジョブとし、ファイルの回復、再割り当てを行った後、再実行指示を行う。



【特許請求の範囲】

【請求項1】 計算機システムでジョブ群を複数回実行する場合、該ジョブ群を実行するとき該ジョブ群に含まれるジョブ毎の実行履歴を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群の障害発生時の実行履歴と正常終了時の実行履歴を比較し、該比較結果に基づき障害の原因を検出することを特徴とする計算機システムの障害回復支援方法。

【請求項2】 請求項1記載の計算機システムの障害回復支援方法において、

前記ジョブ毎の実行履歴としてジョブの終了コードを蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の終了コードを比較し、終了コードが異なるジョブを障害の原因として検出することを特徴とする計算機システムの障害回復支援方法。

【請求項3】 請求項1記載の計算機システムの障害回復支援方法において、

前記ジョブ毎の実行履歴としてジョブにおける入出力回数を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の入出力回数を比較し、実行したジョブのうち入出力回数が正常終了時の最高値と最低値の範囲外にあるジョブを障害の原因として検出することを特徴とする計算機システムの障害回復支援方法。

【請求項4】 請求項1記載の計算機システムの障害回復支援方法において、

前記ジョブ毎の実行履歴としてジョブの実行時間を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の実行時間を比較し、実行したジョブのうち実行時間が正常終了時の最高値と最低値の範囲外にあるジョブを障害の原因として検出することを特徴とする計算機システムの障害回復支援方法。

【請求項5】 請求項1記載の計算機システムの障害回復支援方法において、

前記ジョブ毎の実行履歴としてジョブの実行時間と入出力回数を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の入出力1回当りの実行時間を比較し、実行したジョブのうち入出力1回当りの実行時間が正常終了時の最高値と最低値の範囲外にあるジョブを障害の原因として検出することを特徴とする計算機システムの障害回復支援方法。

【請求項6】 請求項1記載の計算機システムの障害回復支援方法において、

前記ジョブ群の障害発生時の実行履歴と正常終了時の実行履歴を比較し、正常終了時の実行履歴が障害発生時の実行履歴と一致しないジョブを障害の原因として検出することを特徴とする計算機システムの障害回復支援方

法。

【請求項7】 請求項1記載の計算機システムの障害回復支援方法において、

前記ジョブ毎の実行履歴としてジョブの実行時間と入出力回数を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の入出力1回当りの実行時間を比較し、一致しないジョブを障害の原因として検出することを特徴とする計算機システムの障害回復支援方法。

10 【請求項8】 請求項1記載の計算機システムの障害回復支援方法において、

障害が発生したジョブを実行する条件となった先行ジョブの正常終了時の実行履歴と障害発生時の実行履歴を比較して一致しないジョブを障害の原因として検出することを特徴とする計算機システムの障害回復支援方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、計算機システムにおいてジョブ群の実行を複数回繰り返す場合において、ジョブが異常終了したときの計算機システムの回復支援方法に関する。

【0002】

【従来の技術】従来、大規模なジョブネット（多数のジョブの実行順序を規定して順次実行するジョブ群）の監視は、監視用端末装置にジョブネットを表示して行なう方法が知られている（はいたっく1995.11 pp.1-4）。しかし、障害発生後の回復は、人手により、経験的な回復と実行結果を順次出力して原因を検出する方法が取られていた。

【0003】

【発明が解決しようとする課題】上記従来技術では、障害回復は人手による原因解析が必要であり、回復作業者の経験、直感によるところが多いため、回復時間に多量の時間を費やすことがあった。そこで、本発明の目的は、回復時間の短縮を狙い、障害の原因となったジョブを見つけたし、障害回復の支援を行なうことにある。

【0004】

【課題を解決するための手段】上記目的を達成するため、本発明は、計算機システムでジョブ群を複数回実行する場合、該ジョブ群を実行するとき該ジョブ群に含まれるジョブ毎の実行履歴を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群の障害発生時の実行履歴と正常終了時の実行履歴を比較し、該比較結果に基づき障害の原因を検出するようにしている。また、前記ジョブ毎の実行履歴としてジョブの終了コードを蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の終了コードを比較し、終了コードが異なるジョブを障害の原因として検出するようにしている。また、前記ジョブ毎の実行履歴としてジョブにおける入出

力回数を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の入出力回数を比較し、実行したジョブのうち入出力回数が正常終了時の最高値と最低値の範囲外にあるジョブを障害の原因として検出するようにしている。また、前記ジョブ毎の実行履歴としてジョブの実行時間を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の実行時間を比較し、実行したジョブのうち実行時間が正常終了時の最高値と最低値の範囲外にあるジョブを障害の原因として検出するようにしている。また、前記ジョブ毎の実行履歴としてジョブの実行時間と入出力回数を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の入出力1回当たりの実行時間を比較し、実行したジョブのうち入出力1回当たりの実行時間が正常終了時の最高値と最低値の範囲外にあるジョブを障害の原因として検出するようにしている。また、前記ジョブ群の障害発生時の実行履歴と正常終了時の実行履歴を比較し、正常終了時の実行履歴が障害発生時の実行履歴と一致しないジョブを障害の原因として検出するようにしている。また、前記ジョブ毎の実行履歴としてジョブの実行時間と入出力回数を蓄積し、該ジョブ群のジョブ実行において障害が発生した際に、該ジョブ群のジョブについて障害発生時と正常終了時の入出力1回当たりの実行時間を比較し、一致しないジョブを障害の原因として検出するようにしている。また、障害が発生したジョブを実行する条件となった先行ジョブの正常終了時の実行履歴と障害発生時の実行履歴を比較して一致しないジョブを障害の原因として検出するようにしている。

【0005】

【発明の実施の形態】本発明の実施の形態を図1等により説明する。図1は、本発明の実施形態の構成である。図1において、符号100は、ジョブ群106を実行する大型計算機、符号120は、障害回復支援を行なうパーソナルコンピュータ（以下パソコンと記す）、符号130は、大型計算機101とパソコン120を接続するEthernetである。符号111は、ジョブ実行制御102が制御するジョブ群106のJCL(Job Control Language)を格納したファイル、符号112は、ジョブ群106の実行順序を定義したファイル、である。符号101、102、103、104、105、106、107は、大型計算機100で実行するプログラムである。符号101は、大型計算機100で動作するOS(Operating System)、符号102は、ジョブ群106の実行を制御するジョブ実行制御である。符号103は、ジョブ106のファイル入出力取得処理である。符号104は、スプールファイル110に格納されるジョブ実行結果を取り出すジョブ実行結果取り出し処理である。符号10

5は、JCL群111の情報とジョブフロー定義情報112とファイル入出力取得処理103が取得した入出力回数をパソコン120に転送するファイル転送処理である。符号106は、ジョブ群、符号107は、ジョブ実行制御102と入出力回数取得処理103から呼び出されイベントを発生する処理である。

【0006】パソコン120は、符号121、122、123、124、125、126、127で構成する。符号121はメモリ、符号122はディスク装置、符号123は命令を実行するCPU、符号125はディスプレイ装置、符号126はキーボード、符号127はマウスである。メモリ121、ディスク装置122、CPU123、ディスプレイ装置125、キーボード126、マウス127は、内部バス124で接続されている。符号151、152、153は、パソコン120のメモリ121にローディングされCPU123で実行するプログラムである。符号151はジョブ群106を監視する監視処理、符号152はジョブ106で障害発生時に起動する回復支援処理、符号153は監視処理151と回復支援処理152を制御するOSである。符号161、162、163、164はディスク装置122に格納するデータである。符号161は、ジョブDB(Database)であり、JCLと、イベント発生処理107で発生されたジョブ終了イベントおよびクローズイベントを基に作成される。符号162は、ジョブフロー定義情報112をファイル転送105でパソコン120にダウンロードして、表示情報を付加したデータである。符号163は、監視処理151および回復支援処理152で採取したジョブ群106の実行履歴である。符号164は、標準履歴である。標準履歴164の内容はジョブ群106が正常に動作した際の実行履歴である。各データの形式は後で記す。

【0007】以下に、図1を用いて本発明の動作の概要を示した後、図2以降の図を用いて詳細な動作を説明する。本実施形態では、あらかじめ、ジョブフロー定義情報112、JCL群111をファイル転送105によりパソコン120に転送し、ディスク装置122にジョブDB161、ジョブフロー162の形式で格納しておく。ジョブ実行制御は、ジョブをジョブフロー定義情報に従って実行する。具体的には、次の手順で実施する。大型計算機100で動作するジョブ実行制御102は、ジョブフロー定義情報112から実行するジョブ名称を読み込み、JCL群111から実行するジョブのJCLを取り出し、ジョブを起動した後、イベント発生107を呼出し、実行開始イベントをパソコン120の監視処理151に送る。さらに、ジョブ実行制御102は、ジョブが終了すると同様に、実行終了イベントを監視処理151に送る。さらに、大型計算機100では、ジョブ群106のクローズ処理の延長で入出力回数取得処理103が呼び出され、イベント発生107によりクローズ

イベントをパソコン120の監視処理151に転送する。パソコン120の監視処理151は、実行開始イベント、実行終了イベント、クローズイベントを受け取ると、実行開始時刻、実行終了時刻、ファイル入出力回数を実行履歴163に設定する。各イベントの形式は後に記す。

【0008】ここで、大型計算機100で動作するジョブ群106で障害が発生した場合、監視処理151は、回復支援処理152を呼び出す。回復支援処理152は、まず、ディスク装置122の実行履歴163と標準履歴164を比較して障害の原因となったジョブの候補を検出する。検出方法は後で説明する。次に、ジョブDB161とジョブフロー162から障害の原因となった障害原因候補ジョブと障害の発生した障害発生ジョブとの関連を調査し、障害の原因ジョブを検出する。さらに、回復支援処理152は、検出した障害の原因ジョブとジョブルート（障害原因候補ジョブと障害発生ジョブ間のジョブと関連）、をディスプレイ装置125に表示する。操作者は、表示したジョブの入出力ファイルの再割当などの回復処理を行なった後、キーボード126とマウス127で選択してジョブを再実行する。操作者の指示に応じてジョブDB161を参照して、ジョブがアクセスしたファイルの一覧表とファイルをアクセスしたジョブの一覧表の表示も行なう。

【0009】図2は実行履歴163の構成である。図2の符号200は、ジョブ名称を格納するジョブ名称フィールド、符号210はジョブ名称200の終了コードフィールド、符号220はジョブ名称200の開始時刻フィールド、符号230はジョブ名称200の終了時刻フィールド、符号240はジョブ名称200の実行時間フィールド、但し、実行時間＝開始時刻－終了時刻、符号250は、ジョブ名称200のファイル入出力履歴フィールド、符号260は、回復支援処理152がセットする異常フラグフィールドである。ファイル入出力履歴フィールド250は、ジョブ名称200で示されるジョブがアクセスしたファイルのファイル名称フィールド251、ジョブ名称200で示されるジョブがファイル名称251で示されるファイルから入力した回数を格納する入力回数フィールド252、ジョブ名称200で示されるジョブがファイル名称251にで示されるファイルから出力した回数を格納する出力回数フィールド253で構成される。符号261は、入出力回数ビット、符号262は、実行時間ビット、符号263は、終了コードビットである。符号261、262、263の意味と用途は後で図10を用いて記す。上記のジョブ名称200フィールドないしファイル入出力履歴フィールド250に格納されるデータは、後述する図8の監視処理151により得られる。また、異常フラグフィールドのデータは、図9の回復支援処理におけるステップ901の処理、すなわち図10に示す処理によって得られる。

【0010】図3は、ジョブフロー定義情報112の構成である。ジョブフロー定義情報は、ジョブ定義310と先行ジョブ定義320からなる。ジョブ定義310は、識別子フィールド311、ジョブ名フィールド312、からなる。識別子フィールド311は、当該レコードがジョブ定義であることを示す文字列 MJJO を格納する。ジョブ名フィールド312は、ジョブ実行制御102が制御するジョブ名を格納する。先行ジョブ定義320は、識別子フィールド321、ジョブ名フィールド322、先行ジョブ名フィールド323、からなる。識別子フィールド321は、当該レコードがジョブ定義であることを示す文字列 MJWO を格納する。ジョブ名フィールド322は、先行ジョブ名フィールド323に格納したジョブ終了時に起動するジョブ名を格納する。図3の例は、次の通りジョブ実行制御102がジョブを制御することを示す。

【0011】(1) ジョブ1からジョブ4の4件のジョブを制御する。

(2) ジョブ1とジョブ2が終了した時ジョブ3を起動する。

(3) ジョブ3が終了したときジョブ4を起動する。

【0012】図4は、ジョブフロー162の構成を示す図である。ジョブフロー162には、ジョブ毎に図4のテーブルを格納する。図4の符号410は、ジョブ番号である。ジョブ番号410は、ジョブフロー定義情報112内に定義したジョブの通し番号である。符号420はジョブ名称、符号430はジョブを示すノードの配置位置であり、符号431はX座標、符号432はY座標である。440はジョブの状態（実行待、実行中、実行終了）を示す。この状態は色表示にしてもよい。符号450は先行ジョブリスト、符号460は後続ジョブリストである。先行ジョブリスト450はジョブ数451、ジョブ番号452で構成する。同様に後続ジョブリスト460は、ジョブ数461、ジョブ番号462で構成する。図4の例は、図3に示した例のジョブ3のテーブルである。先行ジョブはジョブ番号が001と002の2件であり、後続ジョブはジョブ番号が004の1件であることを示す。ジョブフロー162は、ジョブフロー定義情報112に基づき作成される。

【0013】図5は、ジョブDB161の構成を示す。ジョブDB161は、ジョブ毎に作成するジョブ情報500とファイル毎に作成するファイル情報550からなる。ジョブ情報500は、図4のジョブ番号410を格納するジョブ番号フィールド501、図4のジョブ名称420を格納するジョブ名称フィールド502、当該ジョブのジョブステップの数を格納するステップ数フィールド503、およびステップ数フィールド503に格納した数のステップ情報510からなる。ステップ情報510は、ステップ名称フィールド511、当該ステップで参照するファイルの数を格納するファイル数フィール

ド512、ファイル数フィールド512に格納した数の入出力ファイル情報520からなる。入出力ファイル情報520は、当該ファイルのdd名称フィールド521、ファイル番号フィールド522、入出力種別フィールド523、エラーフラグフィールド524からなる。ファイル情報550は、ファイル番号フィールド551、ファイル名称フィールド552、ボリューム通し番号フィールド553、入出力ジョブ数フィールド554、および入出力ジョブ数フィールドに格納した入出力ジョブ数の入出力ジョブ情報560からなる。入出力ジョブ情報560は、ジョブ名称フィールド561、時刻フィールド562、入出力種別フィールド563からなる。上記ジョブ情報500はJCLを基に作成され、但し、エラーフラグは図7のジョブ終了イベント721における終了コードを基に作成される。また、ファイル情報550はJCLを基に作成され、但し、入出力ジョブ情報560は図7のクローズイベントを基に作成される。

【0014】図6は、入出力回数取得処理103の処理フローである。図6の符号601、602、603は、ジョブ群106の各ジョブのファイル入出力処理の処理フローを示す。ジョブ群106がファイル入出力処理を実施する場合、まず、処理601でファイルのオープンを行ない、処理602でファイル入出力を行なう。ファイル入出力が終了すると処理603によりファイルのクローズ処理を呼び出す。本実施例では、ファイルのクローズ処理の延長で呼び出されるユーザ出口ルーチンに入出力回数取得処理を追加する。具体的には、通常のクローズ処理611終了後、入出力情報取得処理612とイベント発生呼出し処理613を行なう。

【0015】図7に、呼出されたイベント発生107により発生されるイベントの種類と形式を示す。イベントの種類は図7に示す3種類である。符号710はジョブ開始イベント、符号720はジョブ終了イベント、符号730はクローズイベントである。ジョブ開始イベント710は、ジョブ群106のジョブの実行開始時にジョブ実行制御102がイベント発生107を呼出しパソコン120に出力される。ジョブ終了イベント720は、同様に、ジョブ群106の実行終了時にパソコン120に出力される。クローズイベント730は、ジョブ群106のファイル・クローズ処理でイベント発生107を呼出すことによりパソコン120に出力される。ジョブ開始イベント710の符号711は、イベントの識別子、符号712は、実行開始したジョブ名、符号713は、実行開始時刻である。ジョブ終了イベント720の符号712はイベントの識別子、符号722は実行終了したジョブ名、符号723は実行終了時刻、符号724は終了コードである。クローズイベント730の符号731はイベントの識別子、符号732はファイルをクローズしたジョブの名称、符号733はクローズした時

刻、符号734はクローズしたファイル名称、符号735は入力回数、符号736は出力回数である。

【0016】図8は、監視処理151の処理フローである。監視処理151は起動後、処理ステップ801でイベントの発生と終了指示を待つ。終了指示があると処理を終了する。イベントが発生すると、処理ステップ802に制御を移し、イベントの種別を調べる。イベントが、ジョブ開始イベント710であれば処理803へ、ジョブ終了イベント720であれば処理804へ、クローズイベント730であれば処理805へ制御を移す。処理803では、開始時刻712を図2の開始時刻フィールド220に格納する。処理804は、終了時刻723を図2の終了時刻フィールド230へ、終了コード724を終了コードフィールド210へ各々格納する。処理805は、ファイル名734と入力回数735と出力回数736を入出力履歴250へ格納する。イベントがジョブ終了イベントの場合、処理804で次の式により実行時間を算出し、算出した実行時間を実行時間フィールド240に格納し、処理806に制御を移す。実行時間 = 終了時刻 - 開始時刻
処理806は、終了コード724（図7）によりジョブにおける障害の有無を検査する。障害が発生していれば、このジョブを障害発生ジョブとし、処理807によりジョブの実行履歴163を取得して、処理810により障害回復支援を起動する。処理803、805、810終了後再び処理801でイベントの発生と終了指示を待つ。

【0017】図9は、回復支援処理152の処理フローである。回復支援処理152は、まず、ジョブ実行制御102が制御するジョブに対して処理901から処理903を実施する。処理901は障害原因候補ジョブの検出である。検出方法は、図10を用いて後で記す。処理901により検出したジョブは、候補ジョブテーブル（図示省略）に登録する。次に、処理902で障害原因候補ジョブが障害発生ジョブが存在するネット上にあるか否か検査して、無い場合、処理903により該当ジョブを候補ジョブテーブルから削除して、障害原因候補から除く。次に、処理904により障害原因候補ジョブをディスプレイ装置125に表示した後、処理905により指示を待つ。操作者は、キーボード126あるいはマウス127によりファイル一覧表示指示、ジョブ一覧表示指示あるいは再実行指示を行う。ファイル一覧表示指示の場合、処理906により、図5のジョブDB161のジョブ情報500を参照して、当該ジョブの入出力ファイル情報をディスプレイ装置125に表示する。同様に、ジョブ一覧表示指示の場合、処理907により、図5のジョブDB161のファイル情報550を参照して、ファイルをアクセスするジョブ情報の一覧をディスプレイ装置125に表示する。操作者は、ファイル一覧表示906とジョブ一覧表示907の表示を参照して、ファイルの回復や再割当てを行った後、再実行指示を行

う。回復支援処理152は、再実行指示を受け取ると、処理908によりジョブネットに従い再実行コマンドを大型計算機100に投入し、障害ジョブと障害原因候補ジョブ間のジョブを順次再実行する。

【0018】図10は、図9の符号901処理の詳細フローである。本処理は、実行が終了した全てのジョブに対して実施する。処理1001で標準履歴と実行履歴を比較し、その結果により次の処理を実施する。処理1002は、入出力回数が標準範囲内か否か検査する。標準範囲外の場合、処理1003で図2の実行履歴163の異常フラグ260の入出力回数ビット261へ1をセットする。次に処理1004で実行時間が標準範囲内か否か検査して、標準範囲外の場合、処理1005で異常フラグ260の実行時間ビット262へ1をセットする。次に処理1006で終了コードが一致するか否か検査して、不一致の場合、異常フラグ260の終了コードビット263へ1をセットする。最後に処理1008で異常フラグを検査して異常があれば処理1009により候補ジョブテーブルに異常フラグを出力する。

【0019】なお、障害原因候補ジョブを検出する際に、ジョブ毎の実行履歴としてのジョブの実行時間と入出力回数に基づき入出力1回当りの実行時間を求め、ジョブ群のジョブについて障害発生時と正常終了時の上記求めた入出力1回当りの実行時間を比較し、実行したジョブのうち入出力1回当りの実行時間が正常終了時の最高値と最低値の範囲外にあるジョブを障害の原因とするようにしてもよい。また、障害原因候補ジョブを検出する際に、ジョブ毎の実行履歴としてのジョブの実行時間と入出力回数に基づき入出力1回当りの実行時間を求め、ジョブ群のジョブについて障害発生時と正常終了時の上記求めた入出力1回当りの実行時間を比較し、一致しないジョブを障害の原因とするようにしてもよい。さらに、障害が発生したジョブを実行するための条件となった先行ジョブでの正常終了時の実行履歴と障害発生時の実行履歴を比較し、これら両実行履歴が一致しないとき、該先行ジョブを障害の原因とするようにしてもよい。

【0020】最後に、図11～図13を用いて障害発生後の回復手順の具体例を説明する。図11(a)は、本具体例で扱うジョブフローの概念図である。符号1101、1102、1103、1104はジョブ群を、符号1125、1126、1127はジョブルートを示す。ジョブ1101、1102、1103、1104を結ぶ実線はジョブの実行順序を示す。この例では、ジョブA1101とジョブB1102の実行終了時にジョブC1103が実行を開始し、ジョブB1102とジョブC1103の実行終了時にジョブD1104が実行を開始することを示す。このジョブフローは図3のジョブフロー定義で定義する。図11(b)は、図11(a)に示すジョブと各ジョブがアクセスするファイルの関連を示す

図である。符号1111、1112、1113、1114は、ファイルを示す。この例では、ジョブA1101は、ファイルa1111に出力し、同様に、ジョブB1102は、符号1112のファイルb1と符号1113のファイルb2に出力する。ジョブC1103は、符号1111のファイルaと符号1112のファイルb1を入力し符号1114のファイルcを出力する。ジョブD1104は、符号1113のファイルb1と符号1114のファイルcを入力する。このジョブとファイルの関連は図5のジョブDBで定義する。

【0021】図12は、図11に示すジョブの実行履歴の例である。符号1201は実行履歴、符号1202、1203、1204、1205は、各々ジョブA、ジョブB、ジョブC、ジョブDの入出力履歴の詳細である。ここで、具体例では(1)ジョブD実行時に障害が発生し、(2)ジョブA、B、C、D、各々の標準履歴は、図13に示す通り、(3)ファイル入出力回数の標準範囲は±10回、とする。本実施例では、ジョブDの障害を図7の符号720のジョブ終了イベントにより、図8の符号806で認識し、符号811で図9に処理フローを示す障害回復支援を起動する。障害回復支援は、実行履歴と標準履歴を比較して障害原因ジョブを表示する。ジョブA、B、Dの実行履歴の終了コードが標準履歴と異なり、かつ、障害が発生したジョブDへのジョブルート上のジョブであり、さらに、ジョブBのファイルb2に対する入出力回数が、標準範囲外であるのでジョブA、B、Dが障害原因候補ジョブであると判断し、ディスプレイに表示する。

【0022】操作者は、ジョブBが終了コードと入出力回数で標準履歴と異なっているため、ジョブBがアクセスしたファイル一覧の表示指示を行ない、ジョブBがファイルb1とファイルb2に出力したことを知る。次に、入出力回数が標準範囲外であったファイルb2をアクセスしたジョブ一覧の表示指示を行い、ジョブDがファイルb2を入力することを知る。これにより、ジョブDで発生した障害の原因はジョブBであると判断し、障害の回復にはジョブBとジョブDの再実行が必要であると分かる。そこで、操作者はジョブBの処理を確認し、ファイルb1とファイルb2あるいはファイルb2を再割当てしてジョブBとジョブDの再実行指示を行う。

【0023】以上で、障害回復が完了する。

【0024】

【発明の効果】本発明によれば、大規模なジョブネットで障害が発生したときに障害の原因となったジョブを容易に検出できる効果がある。さらに、ジョブがアクセスしたファイルの一覧、ファイルをアクセスしたジョブの一覧を表示することにより、ジョブを再実行する際に回復に必要なファイルを容易に検出できる効果がある。

【図面の簡単な説明】

【図1】本発明の実施例の構成を示すブロック図であ

る。

【図2】ジョブの実行履歴の構成を示す図である。

【図3】ジョブの実行制御を行なうジョブフロー定義情報の構成を示す図である。

【図4】ジョブの監視に使用するジョブフローのファイルの構成を示す図である。

【図5】ジョブDBの構成を示す図である。

【図6】ジョブごとのファイル入出力回数を取得する処理フローを示す図である。

【図7】ジョブを実行する計算機システムからジョブを監視する計算機システムに送るイベントの形式を示す図である。

【図8】ジョブの監視処理の処理フローを示す図である。

【図9】障害の回復支援処理の処理フローを示す図である。

【図10】障害の回復支援処理の障害原因候補ジョブの検出処理のフローを示す図である。

【図11】ジョブフローの具体例の概念図である。

【図12】図11に示すジョブの実行履歴の例である。

【図13】図11に示すジョブの標準履歴の例である。

【符号の説明】

100 大型計算機

101 オペレーティングシステム(OS)

102 ジョブ実行制御処

* 103 入出力回数取得処理

104 実行結果取り出し処理

105 ファイル転送処理

106 ジョブ群

107 イベント発生処理

111 JCL群

112 ジョブフロー定義情報

113 スプールファイル

120 パーソナルコンピュータ

121 メモリ

122 ディスク装置

123 CPU

124 内部バス

125 ディスプレイ装置

126 キーボード

127 マウス

151 監視処理

152 回復支援処理

153 オペレーティングシステム(OS)

161 ジョブDB

162 ジョブフロー

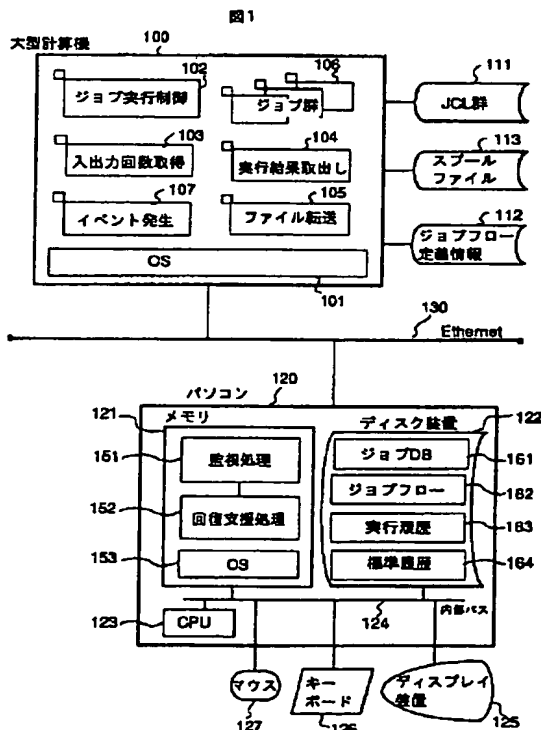
163 実行履歴

164 標準履歴

130 イーサネット(Ethernet)

*

【図1】



【図2】

図2

200	ジョブ名称	ジョブ1	ジョブ2	...
210	終了コード			
220	開始時刻			
230	終了時刻			
240	実行時間			
250	入出力履歴			
260	異常フラグ			

251	ファイル名称	ファイルa	ファイルb	...
252	入力回数			
253	出力回数			

261	入出力回数ビット
262	実行時間ビット
263	終了コードビット

【図3】

図3

識別子	ジョブ名
MJW0	ジョブ1
MJW0	ジョブ2
MJW0	ジョブ3
MJW0	ジョブ4

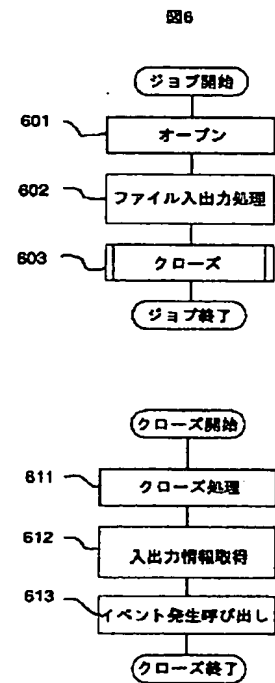
識別子	ジョブ名	先行ジョブ名
MJW0	ジョブ3	ジョブ1,ジョブ2
MJW0	ジョブ4	ジョブ3

【図4】

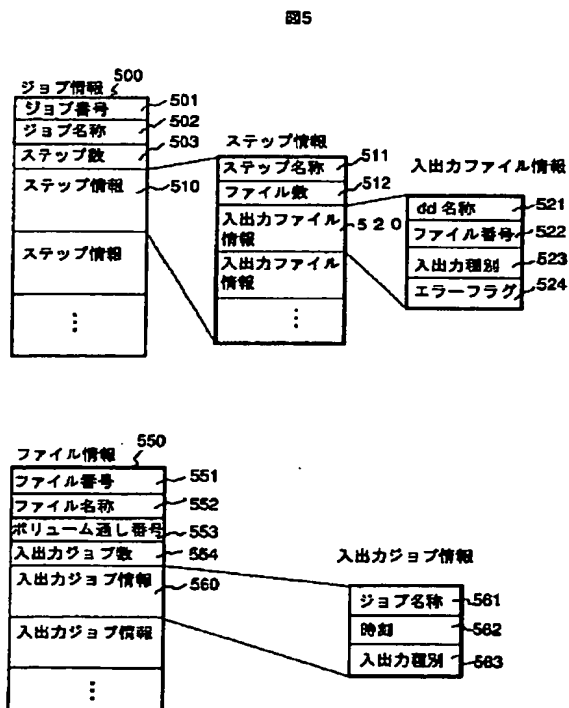
図4

410	ジョブ番号	003
420	ジョブ名称	
430	配線	X
430	位置	Y
440	状態	
450	先行ジョブリスト	ジョブ数 2 ジョブ番号(1) 001 ジョブ番号(2) 002
460	後続ジョブリスト	ジョブ数 1 ジョブ番号(1) 004

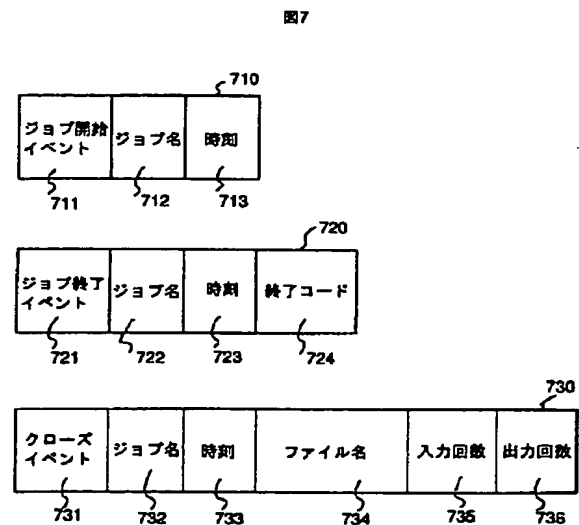
【図6】



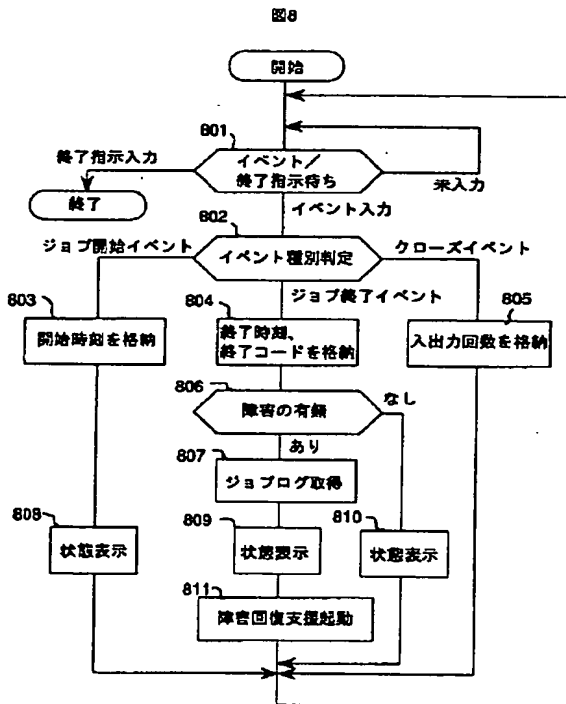
【図5】



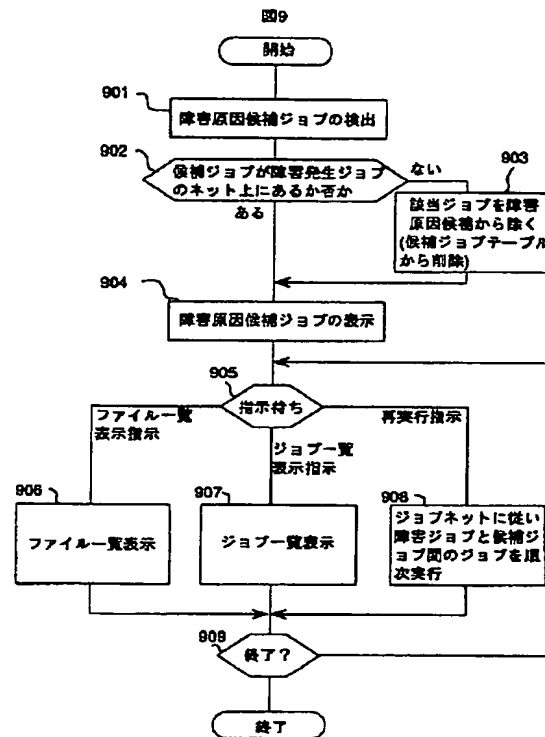
【図7】



【図8】

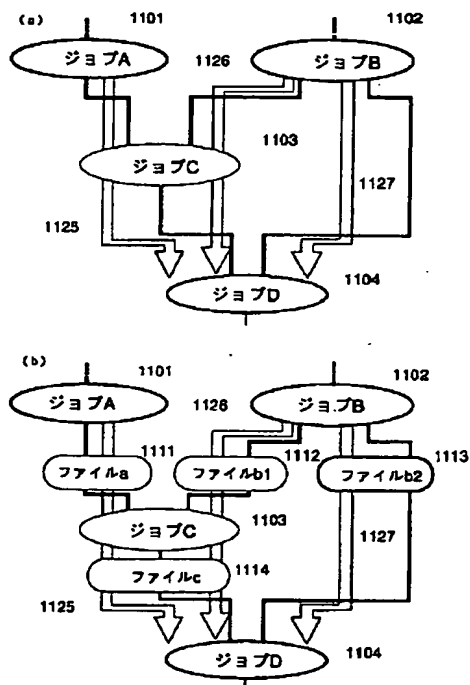


【図9】

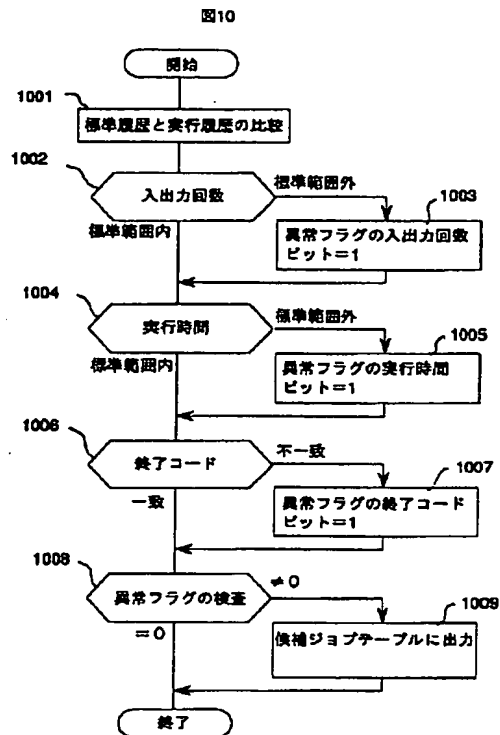


【図11】

図11



【図10】



【図12】

図12

1201	ジョブ名称	ジョブA	ジョブB	ジョブC	ジョブD
	終了コード	4	4	0	16
	開始時刻	10:00	10:00	10:10	10:15
	終了時刻	10:10	10:05	10:15	10:20
	実行時間	600秒	300秒	300秒	300秒
	入出力履歴				
	異常フラグ	000	000	000	000

ジョブA

1202	ファイル名称	ファイルa
	入力回数	
	出力回数	100

ジョブB

1203	ファイル名称	ファイルb1	ファイルb2
	入力回数		
	出力回数	100	10

ジョブC

1204	ファイル名称	ファイルa	ファイルb1	ファイルc
	入力回数	100	100	
	出力回数			50

ジョブD

1205	ファイル名称	ファイルb2	ファイルc
	入力回数	10	50
	出力回数		

【図13】

図13

1301	ジョブ名称	ジョブA	ジョブB	ジョブC	ジョブD
	終了コード	0	0	0	0
	開始時刻	10:00	10:00	10:10	10:15
	終了時刻	10:10	10:05	10:15	10:20
	実行時間	600秒	300秒	300秒	300秒
	入出力履歴				
	異常フラグ	000	000	000	000

ジョブA

1302	ファイル名称	ファイルa
	入力回数	
	出力回数	100

ジョブB

1303	ファイル名称	ファイルb1	ファイルb2
	入力回数		
	出力回数	100	50

ジョブC

1304	ファイル名称	ファイルa	ファイルb1	ファイルc
	入力回数	100	100	
	出力回数			50

ジョブD

1305	ファイル名称	ファイルb2	ファイルc
	入力回数	10	50
	出力回数		

フロントページの続き

(72)発明者 廣澤 敏夫

東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.